

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Факультет електроніки та інформаційних технологій

Кафедра електроніки,
загальної та прикладної фізики

Кваліфікаційна робота магістра
**РЕАЛІЗАЦІЯ МІКРОКОНТРОЛЕРНИХ СИСТЕМ МОНІТОРИНГУ ТА
КЕРУВАННЯ НА БАЗІ WEB СЕРВЕРІВ**

Виконав
студент групи ЕП.м – 81н



В. С. Лобода

Науковий керівник,
к. ф.-м. н., ст.викл.



К. В. Тищенко

Завідувач кафедри ЕЗПФ
д-р фіз.-мат. наук, професор



І. Ю. Проценко

Суми 2020

РЕФЕРАТ

Об'єктом дослідження науково-дослідної роботи є Web-сервери та їх прикладне використання в системах моніторингу і керування.

Мета роботи полягає у вивченні принципів роботи та особливостей реалізації інформаційних систем на базі Web-серверів; Реалізації функціоналу сервера на базі апаратних мікроконтролерних платформ.

У роботі використовувався пристрій Node-MCU з вбудованим контролером ESP-8266. Розглянуто приклади серверів, що надають інформаційні послуги, показано їх основні особливості та відмінності. У результаті проведених досліджень встановлено, що для використання в системах малої автоматизації та інтернеті речей доцільно використовувати мікроконтролери з вбудованими мережевими інтерфейсами, наприклад, ESP-8266, завдяки його відносній доступності та простоті програмування.

На основі платформи ESP-8266 було реалізовано сервери із використанням стандартних засобів розробки та середовища Arduino IDE для керування та моніторингу в межах інформаційних систем типу «інтернет речей», зроблено порівняння їх функціоналу та складності розробки при кожному із підходів до розробки.

Робота викладена на 33 сторінках, зокрема містить 11 рисунків, список цитованої літератури із 21 джерела.

КЛЮЧОВІ СЛОВА: WEB-СЕРВЕР, МІКРОКОНТРОЛЕР, ДАТЧИК, СИСТЕМА МОНІТОРИНГУ, ESP-8266.

ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1. ХАРАКТЕРИСТИКИ ВЕБ-СЕРВЕРІВ ТА	
ВИМИРЮВАЛЬНИХ ПЕРЕТВОРЮВАЧІВ.....	6
1.1 Призначення та основні характеристики веб-серверів	6
1.2 Найпоширеніші сервери та їх установчі пакети	8
1.3 Перетворювачі фізичних величин в системах моніторингу	10
РОЗДІЛ 2. ХАРАКТЕРИСТИКИ ESP8266 ТА СЕРВЕРИ НА ЙОГО	
ОСНОВІ.....	18
2.1 Характеристики мікроконтролерної платформиESP8266	18
2.2 Програмування налагодження і підключення контролера ESP-8266....	18
2.3 Сервер на основі ESP8266.....	21
2.4 Реалізація сервера на основі ESP-8266 із використанням Arduino IDE	26
ВИСНОВКИ	31
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	32

ВСТУП

Веб-сервер – сервер, що приймає HTTP-запити від клієнтів, зазвичай веб-браузерів, і видає їм HTTP-відповіді, як правило, разом з HTML-сторінкою, зображенням, файлом, медіа-потокком або іншими даними.

Веб-сервером називають як програмне забезпечення, яке виконує функції серверного обладнання, так і безпосередньо комплекс апаратного забезпечення, на якому дане програмне забезпечення працює.

Клієнт, яким зазвичай є веб-браузер, відсилає веб-серверу запити на отримання ресурсів, позначених URL-адресами. Ресурси – це HTML-сторінки, зображення, файли, медіа-потоки або інші дані, які необхідні клієнту. У відповідь веб-сервер передає клієнту запитані дані. Цей обмін відбувається за протоколом HTTP або HTTPS.

Мета кваліфікаційної роботи магістра полягає у огляді сучасних систем управління web-серверами, а також розробці схеми та програмного забезпечення сервера на базі мікроконтролера ESP8266, який забезпечує віддалений доступ до моніторингу фізичних величин, та керування обладнанням в режимі реального часу.

РОЗДІЛ 1

ХАРАКТЕРИСТИКИ ВЕБ-СЕРВЕРІВ ТА ВИМИРЮВАЛЬНИХ ПЕРЕТВОРЮВАЧІВ

1.1 Призначення та основні характеристики веб-серверів

Веб-сервер – це апаратно-програмна платформа, яка функціонально приймає запити від підключеного авторизованого користувача і повертає йому відповідь у вигляді інформаційного потоку, який може містити документи, або сторінки веб сайту. Будь-який комп'ютер можна виступати мережевим сервером, для чого необхідно здійснити його налаштування у відповідності до поставлених цілей, здебільшого потрібно встановити спеціалізовану оболонку із відповідним програмним пакетом, що забезпечує необхідний функціонал. Вимоги до технічного оснащення серверів визначаються числом розміщених на ньому ресурсів та вимогами до пропускної здатності. Чим вони більші, тим потужніше обладнання повинен містити сервер [1].

Для більшої наглядності можна привести аналогію з інформацією в бібліотеці. Ви, як клієнт, заходите до бібліотеки, яку можна представити сервером, і подаєте запит на видачу вам книги. Бібліотекар, який виступає у вигляді оболонки, знаходить необхідну інформацію на сервері і надсилає її вам. Можна подати запит до оболонки на додаткову інформацію – аналогічно переходу за посиланням. Важливо розуміти, що один і той же ресурс в мережі Інтернет може одночасно обслуговувати велику кількість число користувачів, тобто клієнтів.

Обслуговування сервером запитів клієнта відбувається схожим чином: приходячи до бібліотеки за літературою, ви можете задати питання до бібліотекаря, здійснити операцію схожу на пошук в інтернеті, або знайти необхідну інформацію за бібліотечним каталогом. Такий підхід дозволяє швидко знайти потрібну інформацію [1, 2].

Головною задачею веб-сервера є зберігання інформації, такої, як web-

сторінки, файли, графічні зображення, текстова інформація.

Веб-сервер, відповідно до свого функціоналу може виконувати наступні завдання:

- отримання запитів;
- запуск скриптового коду на спеціалізованих мовах програмування у вигляді серверних аплетів;
- формування контенту для веб-сторінки;
- відправка інформації користувачеві, у виді відповіді на запит;
- захист інформації;
- проведення ідентифікації підключених користувачів;
- ведення логів (журналу звернень);
- обслуговування запити інших, не базових типів (mailto, FTP, UDP, P2P та ін.).

Для розуміння логіки роботи веб-сервера, необхідно мати уявлення про принципи комунікації в мережі. У їх основі лежать певні набори правил, так звані протоколи, які описані URL (визначається вказівкою типу – ftp, http://, https:// та ін.) [2 – 4].

HyperTextTransferProtocol– протокол передачі гіпертексту. Сторінки будь-якого сайту в інтернеті завжди представлені у виді гіпертекстового документа [4]. Формування такого документа є кінцевим результатом роботи серверного або клієнтського web-додатка.

Опишемо покроково даний процес [3, 4]:

- користувач вводить адресу сторінки, або переходить за посиланням, браузер відсилає запит, кодуючи інформацію за правилами протоколу, котрий застосовується у конкретній сесії;
- хост, до котрого прикріплена дана мережева адреса, запускає серверні програми. Кожна з цих програм виконує функції, описані для неї;
- отримані від клієнта дані розшифровуються, а отримані команди виконуються;

- формується гіпертекстовий документ, який певним чином кодується і пересилається клієнту;
- браузер клієнта отримує відповідь, розкодовує її в html і відображає на екрані сторінку.

Необхідною умовою для функціонування веб-сервера є наявність апаратного забезпечення (комп'ютера або іншого пристрою), котрий буде займатись обробкою запитів. Вибір апаратної бази сервера залежить від кількості одночасно підключених користувачів і характеру виконуваних ним операцій, тобто чим більше запитів і складніші розрахунки повинен обробляти сервер, тим потужніше обладнання необхідно застосовувати.

Насьогодні існує велика кількість компаній, які надають послуги хостингу, у яких ви орендуєте сервер. Вам дається фізичне місце і квота для розміщення файлів та матеріалів сайту. Але якщо ваш сервер виконує прості функції, то можна використати власне обладнання, налаштувавши попередньо його програмне забезпечення.

Коли підібрано обладнання та програмне забезпечення сервера, необхідно надати йому статичну IP-адресу. Сайт стане доступним на веб-сервері після реєстрації доменного імені та виконання перетворення адрес службою DNS – прив'язування IP-адреси (наприклад, адреси 185.351.166.31) і доменного імені (www.site.com) [4].

1.2 Найпоширеніші сервери та їх установчі пакети

Розглянемо найбільш популярні оболонки систем управління серверами.

Сервер Apache – безкоштовний програмний пакет, що реалізовує функціонал web-сервера, він вільно розповсюджується, оскільки ліцензується відкритою ліцензією, і має масу корисних функцій та переваг перед конкурентами [3], а саме:

- постійна підтримка команди розробників;
- модулі для роботи з серверними мовами програмування PHP, Perl,

Python, Ruby, ASP і т.д.;

- відкритий код, який можна модернізувати і доопрацьовувати під власні потреби, чим активно і займаються різні програмісти;
- кросплатформеність. Спочатку Apache був створений під операційну систему Unix, але згодом почав підтримуватись Windows, MacOS, BSD, Linux, OS/2 і NovellNetWare;
- потужні засоби забезпечення мережевої безпеки.

При інсталяції сервера Apache необхідно вказати ім'я майбутнього хоста. До каталогу htdocs, який розміщений всередині папки Apache.x (де x.x – номер версії) потрібно помістити будь-яку сторінку з розширенням *.html [4].

Коли в каталог буде поміщений файл, потрібно відкрити браузер і ввести адресу: localhost://ім'я_сторінки.html. На екрані з'явиться текст, поміщений до html файлу – це і означає, що сторінка відкрита з налаштованого вами сервера. Якщо ви отримали повідомлення про помилку доступу «Не вдається отримати доступ до сайту», значить із якихось причин не запущений сервер Apache. У такому разі необхідно натиснути на нього і вибрати «Play». Після цього сервер запрацює.

Сервер NGINX. Один із найпопулярніших програмних пакетів керування серверами. Загальна доля активних майданчиків, що ним керуються складає близько 22%. Він здебільшого використовується великими компаніями і професійними розробниками, оскільки витримує велике навантаження запитів на сервер, також він зарекомендував себе, як надійний, безпечний і продуманий. Розповсюджується дана система керування серверами вільно, але існують і платні версії Plus із розширеною підтримкою виробників та додатковим функціоналом [4].

Сервер IIS, який розробляється компанією Microsoft. Популярність даного сервера забезпечена, в основному, ім'ям розробника. Він представляє собою набір веб служб та тісно інтегрований з операційною системою Windows, і не може бути запущений на інших платформах. Основною

платформою програмування у даному сервері є ASP.NET, але існує і альтернатива, наприклад, PHP.

Для повноцінної організації хостингу на даному програмному сервері необхідно встановлення серверної версії операційної системи Microsoft – Windows Server. Даний сервер купується разом з операційною системою і є невід'ємною її частиною.

Установчі пакети

Для зручного керування сервером розроблені інструменти, що дозволяють достатньо просто розгорнути на своєму комп'ютері web-сервер. Розглянемо найбільш популярні з них.

OpenServer – представляє собою портативне середовище розробки, котре містить велику кількість баз даних, популярних мов програмування, а також додаткові сервіси [3]. Наприклад, інтерфейс роботи з БД PhpMyAdmin. На сьогодні OpenServer – це найпопулярніший установочний набір серверного програмного забезпечення, котрий може працювати навіть із знімного носія.

Xampp – підтримує роботу з такими програмними рішеннями, як Apache, Php, Perl, MariaDB тощо. Має власну панель керування, і розповсюджується за вільною ліцензією [3].

Denwer – зручний набір найнеобхідніших інструментів, що включає в себе Apache, PHP, MySQL, PhpMyAdmin. Насьогодні даний проект більше не підтримується, тому версії програм, розміщені у ньому не актуальні.

1.3 Перетворювачі фізичних величин в системах моніторингу

До вимірювальних контролерів можна підключити різноманітні датчики. Датчик – це окремий пристрій, який здатний вимірювати певну фізичну величину або реагувати на фізичне явище і передає інформацію про це в вигляді електричного сигналу [5].

Вимірювальне перетворення – операція перетворення вхідного сигналу у вихідний, інформативний, параметр якого з заданою точністю функціонально

пов'язаний з інформативним параметром вхідного сигналу.

Вимірювальний претворювач (ВП) – це технічний засіб, що реалізує вимірювальне перетворення.

Залежність між інформативними параметрами вхідного і вихідного сигналів ВП встановлюється номінальною статичною характеристикою перетворення, яка може виражатися у вигляді рівняння, графіка або таблиці. Велика частина природних залежностей між фізичними сигналами має нелінійний характер [5].

Чутливий елемент – це вимірювальний перетворювач, що забезпечує перетворення вимірюваної величини в інформативний сигнал, зручний для подальшого використання (передачі, обробки, зберігання тощо). Будь-який первинний вимірювальний перетворювач чутливий до різних фізичних величин. Наприклад, п'єзоелемент формує електричний заряд під впливом таких величин як сила, температура, вологість, оптичне випромінювання [6].

Датчики розрізняються по тому, що вони фіксують. Існують датчики для вимірювання температури, відстані, вібрації, звуку, вологості, руху, магнітного поля та багато інших.

Також вони розрізняються за типом сигналу. Деякі передають сигнал у вигляді змінної напруги (аналогові), деякі у вигляді послідовності низької і високої напруги (цифрові), інші змінюють власний опір. Тип сигналу визначає як фізичні показання проєктуються на електричні величини придатні для зчитування.

Далі, сенсори розрізняються по протоколу. Він визначає як приймаюча сторона (наприклад ESP-8266) повинна інтерпретувати сигнал, щоб отримати значення вимірюваної величини. Для одних сенсорів вимірювана величина прямо пропорційна переданій напрузі, інші передають тільки «так/ні» за допомогою двох величин напруги, треті передають свої показання у вигляді послідовності біт, де відсутності напруги відповідає 0, а високій напрузі - 1. Протокол у кожного сенсора свій і визначається виробником датчика.

Крім того, сенсори розрізняються за способом фізичного підключення.

Велика кількість Arduino-сумісних сенсорів легко підключаються по трьом проводам. Інші сенсори мають контакти для підключення через breadboard.

Щоб з'єднати будь-який сенсор з контролером і почати зчитувати його покази, потрібно:

- визначити якого типу сигналу відповідає сенсор: аналоговий, цифровий, опір;
- фізично підключити сенсор до Arduino. Залежно від сенсора: по трьом проводам або через breadboard;
- визначити за яким протоколом передаються дані: залежність від напруги, бінарний або свій власний, і написати програму для їх отримання [5].

Розглянемо датчики, що можна використати для побудови вимірювальної системи з подальшим підключенням до веб-сервера.

Гігрометр – це прилад для вимірювання вологості. Параметри вологості є важливим параметром як для живих організмів, так і для предметів, що нас оточують: техніки, конструкційних матеріалів та ін. Наприклад, від вологості значною мірою залежить самопочуття та комфортність людей. Деякі прилади здатні працювати лише в рамках певного мікроклімату, тому настройка і калібрування приладів повинна проводитись за точних значень вологості.

Найпростішим гігрометром може виступати конденсатор з повітряним проміжком між обкладинками (рис. 1.1 а). Діелектрична проникність повітря залежить від вологості, а її зміна призведе до зміни ємності такого датчика.

Технологічно складнішим, проте більш надійним буде прилад із конденсатором, не із повітряним зазором, а діелектриком, діелектрична проникність якого значною мірою змінюється від вологості. Таке вдосконалення датчика дозволяє поліпшити його робочі характеристики, і зменшити вплив оточуючого середовища [6]. Також, ємнісний датчик з діелектричним зазором, чутливим до вологи, може бути використаний для вимірювання рівня вологості твердих речовин. Між обкладками конденсатора розміщується об'єкт, вологість котрого необхідно визначити,

конденсатор підключається до генератора і LC коливального контуру. Про величину вологості можна судити із частоти коливання, яка вимірюється частотоміром або лічильником. Такий метод має ряд недоліків: за вологості нижче 0,5% його точність різко знижується, також необхідно проводити очистку поверхні зразка від частинок з високою діелектричною проникністю, крім того значущим фактором є форма зразка, яка не повинна змінюватися в процесі вимірювання.

Насьогодні все більшого поширення набувають тонкоплівкові ємнісні гігрометри (рис. 1.1 б). Такий вимірювальний перетворювач складається з підкладки, на яку нанесені два електроди гребінчастої форми, на рисунку один з них показано червоним, інший синім. Ці два електроди будуть відігравати роль обкладок конденсатора. Для додаткової термокомпенсації в датчики такого типу додаються два датчика температури.

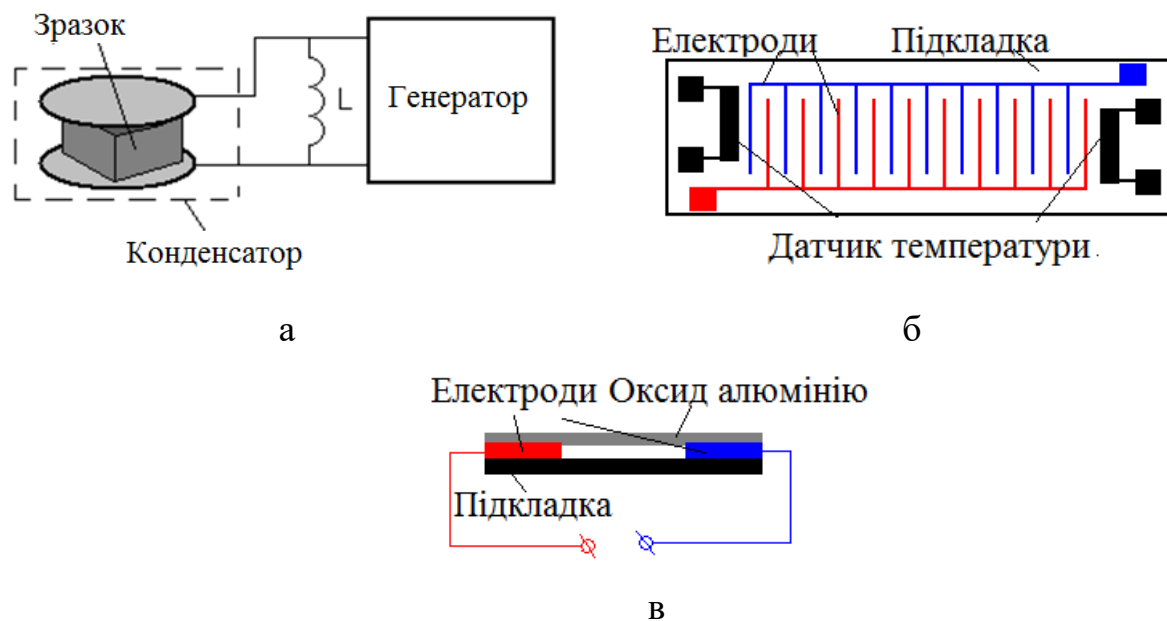
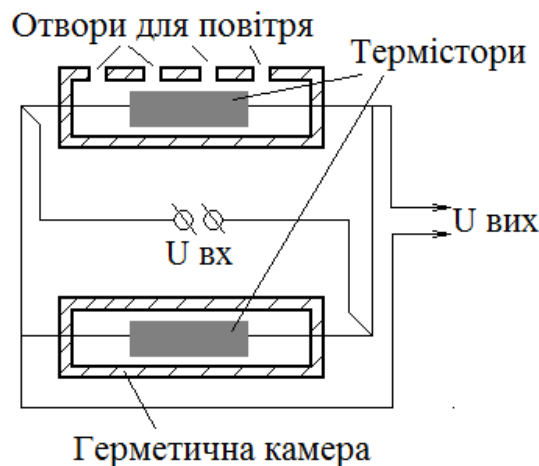


Рис.1.1 Будова ємнісного датчика вологості: на основі масивних електродів (а), тонкоплівкової технології (б) та резистивного датчика (в). Адаптовано з роботи [6]

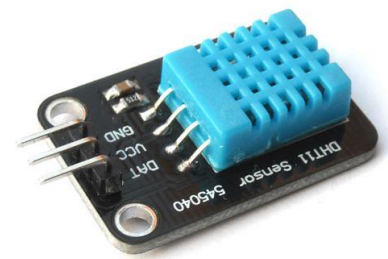
Резистивний датчик (рис. 1.1 в) представляє собою два електроди, що нанесені на підкладку, поверх електродів наноситься поверхневий шар матеріалу, опір якого значною мірою залежить від вологості. Таким

матеріалом може бути, наприклад, пористий оксид алюмінію. Такий матеріал добре поглинає воду з навколишнього середовища, внаслідок чого змінюється його питомий опір, як наслідок загальний опір системи буде залежати від вологості. Головна перевага таких вимірювальних перетворювачів їх мала вартість.

Термісторні гігрометри (рис. 1.2 а) складаються з двох терморезисторів, тобто резисторів, опір яких залежить від зміни температури оточуючого його середовища. Один з цих терморезисторів поміщають до герметичної оболонки, котра наповнена чистим сухим повітрям. Другий датчик поміщається до камери, з'єднаної із атмосферою невідомої вологості.



а



б

Рис.1.2 Будова термісторного датчика вологості (а) [6] та датчика вологості і температури DHT11 (б)). Адаптовано з роботи [7]

Для реалізації датчика із таким чутливими елементами, терморезистори з'єднують у мостову схему. До однієї діагоналі вимірювального моста прикладається напруга, а з іншої знімаються показання. Якщо вихідна напруга рівна нулю, то температура обох термісторів однакова, і, як наслідок, вологість також однакова. У випадку, якщо на виході з'являється напруга, то міст розбалансовується, що свідчить про те, що вологості в сухій і вимірюваній комірках різні. За величиною і знаком напруги можна визначити

про величину вологості. При підвищенні вологості з поверхні терморезистора починає випаровуватися волога, внаслідок чого температура зменшується, чим вище вологість, тим інтенсивніше проходить випаровування і, як наслідок, більше охолоджується терморезистор, і змінюється його опір [6].

Датчик вологості і температури DHT11 (рис. 1.2 б) – це цифровий датчик температури і вологості, що дозволяє отримувати дискретний калібрований сигнал на виході. Даний вимірювальний перетворювач складається з плівкового ємнісного датчика вологості і терморезистора. Також до схеми датчика включений АЦП для перетворення аналогових значень вологості і температури в зручну для використання форму [7].

Газоаналізатор – прилад для визначення якісного і кількісного складу сумішей газів. Існують такі прилади різні за способом вимірювання, але робота всіх таких приладів основана на вимірюванні фізичних, фізико-хімічних параметрів суміші газів, або її окремих компонентів.

Термохімічні газоаналізатори, дія яких заснована на реакції окислення, використовуються для вимірювання концентрації CO, H₂, метану, етилену, парів бензину та інших горючих речовин. Вони поділяються на дві групи: газоаналізатори, де каталізатором, на якому відбувається реакція окислення, є сам термоперетворювач, і газоаналізатори з насипним каталізатором, в яких термоперетворювач служить тільки для вимірювання теплового ефекту.

При нагріванні в повітрі до певної температури полікристала двоокису олова SnO₂ (рис. 1.3) з електронною провідністю кисень адсорбується на поверхню кристала з надмірним негативним зарядом. При цьому надлишкові електрони передаються адсорбованому кисню, утворюючи шар об'ємного позитивного заряду. Таким чином, формується поверхневий потенційний бар'єр, що перешкоджає протіканню струму через зерна полікристалічної структури. Електричний опір датчика визначається висотою цього потенційного бар'єру. У присутності відновлюючого газу поверхнева щільність негативного заряду зменшується з пониженням висоти потенційного бар'єру між зернами що, в свою чергу, зменшує опір датчика.

Адсорбція викликає збільшення або зменшення провідності напівпровідника в залежності від того, який газ (акцепторні або донорний) адсорбується на напівпровіднику (з електронною або дірковою провідністю). Адсорбція кисню на напівпровіднику з електронною провідністю, виготовленому з діоксиду олова, знижує провідність, тоді як метан надає протилежний вплив. Молекули газу, викладені на поверхню, діють подібно атомам домішки, введеним в кристал: вони або збіднюють, або збагачують напівпровідник носіями струму

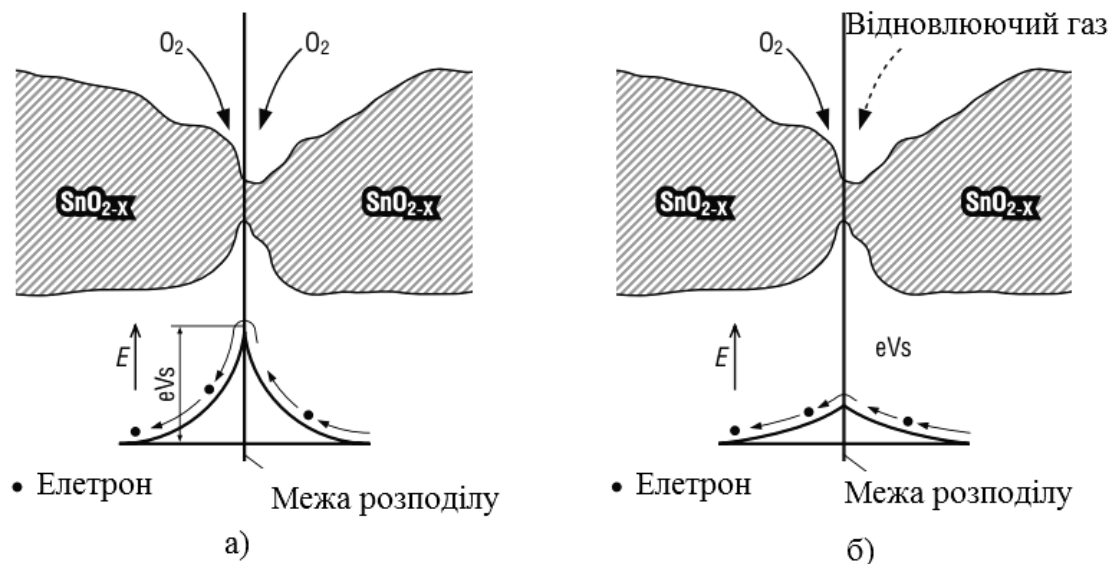


Рис.1.3 Потенційний бар'єр на межі полікристалів в відсутності (а) і в присутності (б) відновлюючого газу). Адаптовано з роботи [5]

Датчик газу MQ-4 (рис. 1.4) конструктивно містить, у якості чутливого елемента, газоаналізатор, який дозволяє детектувати наявність в навколишньому середовищі вуглеводневих газів (бутан, пропан, метан), водень та дим.

На виході датчика формується аналоговий сигнал, величина якого пропорційна вмісту газів, на які реагує газоаналізатор. На платі датчика розміщений резистор змінного опору, яким можна відрегулювати чутливість

вимірювання. До складу газоаналізатора входить нагрівач, який служить каталізатором хімічної реакції.

Для отримання стабільних показань перед початком використання датчик необхідно один раз прогріти протягом щонайменше доби. Після цього стабілізація показань після увімкнення займатиме близько однієї хвилини.



а



б

Рис.1.4 Зовнішній вигляд датчика газу MQ-4 (а) [9] та температури на основі DS18B20 (б) [10]

Напівпровідникові датчики температури призначені для вимірювання температури від -55° до 150° C. В цей діапазон потрапляє величезна кількість завдань, як в побутових, так і в промислових додатках. Завдяки високим характеристикам, простоті застосування і низької вартості напівпровідникові датчики температури виявляються дуже привабливими для застосування в мікропроцесорних пристроях вимірювання та автоматики.

Датчик на основі мікросхеми DS18B20 (рис. 1.5) для перетворення температури в цифрове значення безпосередньо в корпусі датчика дає переваги у вигляді високої точності і стабільності показань, відсутності впливу з'єднувальних проводів на показання датчика, легкості підключення до мікроконтролерів і ПК, можливість роботи декількох датчиків на одному шлейфі.

РОЗДІЛ 2

ХАРАКТЕРИСТИКИ ESP8266 ТА СЕРВЕРИ НА ЙОГО ОСНОВІ

2.1 Характеристики мікроконтролерної платформи ESP8266

ESP8266 представляє собою мікроконтролер з інтегрованим інтерфейсом Wi-Fi виробництва компанії Espressif Systems. Окрім вбудованого модуля Wi-Fi, мікроконтролер характеризується відсутністю флеш-пам'яті в SoC, тому мікрокод контролера виконуються із зовнішньої флеш-пам'яті, підключеної за інтерфейсом SPI [11].

Мікроконтролер набув значної популярності в 2014 році разом з виходом перших продуктів на його основі, а також внаслідок низької вартості. На початку 2016 року розпочалося виробництво контролера ESP8285, що поєднував у собі ESP8266 і флеш-пам'ять на 1 МБайт. Разом із цим Espressif представила логічне продовження лінійки цього сімейства контролерів – мікросхему ESP32 і модулі на її основі.

Як було зазначено вище, мікроконтролер не має на кристалі вбудованої пам'яті. Виконання програми здійснюється із зовнішнього SPI ПЗП шляхом динамічного зчитування необхідних в конкретний момент часу ділянок мікрокоду до кешу інструкцій. Мікроконтролером підтримується до 16 МБ зовнішньої пам'яті програм типів Standard, Dual або Quad SPI.

ESP8266 повністю універсальний, що дозволяє застосовувати його в будь-яких системах, де вам потрібен контроль зв'язку з інтернетом. Іншими словами, якщо необхідно приймати і посилати лише певні пакети по певним протоколам, для фільтрації, захисту системи від злому і DDOS атак, пристрій вам стане в нагоді [12 – 14].

Саме можливість налаштування кількості пакетів для прийому від одного користувача за одиницю часу робить його краще звичайних плат розширення, що надають системам доступ до інтернет-з'єднанню. Користувачі можуть самостійно написати програму під сортування,

відправку і фільтрацію пакетів, без необхідності використовувати вбудовані авторами і неоптимізовані алгоритми без використання ООП. Однак розробники все ж подбали про тих, хто не сильний в програмуванні і у себе на сайті відразу представили список програм відповідних для установки на мікроконтролер в різних ситуаціях.

За своєю суттю, пристрій має велику область застосування, від можливості зібрати найпростіший Wi-Fi-модем і до складних систем, на прикладі розумного будинку. Адже пристрій сам здатний поширювати спійману мережу, стаючи точкою доступу, що також дозволяє застосовувати його в якості підсилювача сигналу і одночасно з тим, фільтра з'єднання. Однак швидкість поки залишає бажати кращого, але завжди можна придбати підсилювач.

2.2 Програмування, налагодження і підключення контролера ESP-8266

Підключити модуль можна за допомогою звичайної макетної плати і з'єднувачів типу Dupont, що підходять до обраної вами плати. За бажанням можете використовувати кабель USB-TTL та інші, які підходять до описаних вище протоколів з'єднання з ПК, і конвертер від PL2303 і до FTDI [14].

Необхідно під'єднати зворотний кінець кабелю в порт комп'ютера і чіп відразу повинен розпізнатися, далі досить завантажити будь-яку утиліту, яка підходить для написання Ардуіно-скриптів або скористатися вже наявною.

Виробником мікроконтролера не надається документація щодо моделі команд внутрішньої периферії. Натомість він розповсюджує набір бібліотек, через з'єднувальний інтерфейс котрих (API) програміст може одержати доступ до периферійних пристроїв. Оскільки дані бібліотеки активно використовують ОЗП контролера, то виробник не вказує точну її кількість на кристалі, а тільки приблизну оцінку кількості ОЗП, що залишиться користувачеві після підвантаження бібліотек, а це близько 50 кБ. Незалежні користувачі, які досліджували бібліотеки ESP8266, стверджують, що він

містить 32 кБ кешу інструкцій і 80 кБ ОЗП даних [15 – 17].

Джерело виконуваної програми ESP8266 задається станом портів GPIO0, GPIO2 і GPIO15 у момент зняття сигналу Reset (фізично це реалізовано подачею живлення на контролер). Найбільш часто використовувані два режими роботи: виконання коду з UART (GPIO0 = 0, GPIO2 = 1 і GPIO15 = 0) та із зовнішнього ПЗП (GPIO0 = 1, GPIO2 = 1 і GPIO15 = 0). Режим виконання коду з UART використовується для перепрошивки підключеної флеш-пам'яті, а другий – це основний режим роботи за замовчуванням [17].

Типове застосування ESP8266 як апаратної основи Internet of Things найчастіше має на увазі установку в будинках або офісах. При цьому підключення до мережі здійснюється до домашньої / офісної локальної мережі з виходом в інтернет через роутер. Власник пристрою може контролювати його за допомогою планшета або комп'ютера через свою локальну мережу або віддалено, через Інтернет.

ESP8266 може працювати як точка доступу так і кінцевий термінал. При нормальному режимі роботи в локальній мережі контролер ESP8266 конфігурується в режим кінцевого терміналу. Для цього пристрою необхідно задати SSID Wi-Fi мережі і, в закритих мережах, пароль доступу. Для початкового конфігурування цих параметрів зручний режим точки доступу. У режимі точки доступу пристрій видно при стандартному пошуку мереж в планшетах і комп'ютерах. Залишається підключитися до пристрою, відкрити HTML сторінку конфігурації і задати параметри мережі. Після чого пристрій штатно підключиться до локальної мережі в режимі кінцевої станції. У разі виключно місцевого використання можливо завжди залишати пристрій в режимі точки доступу, що знижує необхідні зусилля користувача по його настройці [18].

Після підключення до Wi-Fi мережі пристрій повинен отримати IP-параметри локальної мережі. Ці параметри можна задати вручну разом з параметрами Wi-Fi або активізувати будь-які сервіси автоматичної конфігурації IP-параметрів (наприклад, DHCP). Після настройки IP

параметрів звернення до сервера пристрою в локальній мережі зазвичай здійснюється по його IP адресою, мережевим ім'ям (в разі якщо імена підтримані будь-якої технологією, наприклад NBNS) або сервісу (в разі якщо підтриманий автоматичний пошук сервісів, наприклад через протокол SSDP).

Найчастіше доступ до пристрою потрібно з Інтернету. Наприклад користувач з мобільного телефону віддалено перевіряє стан свого «розумного будинку», звертаючись безпосередньо до пристрою. В цьому випадку пристрій працює в режимі сервера, до якого звертається зовнішній клієнт. Як правило, пристрій на основі ESP8266 знаходиться в локальній мережі офісу або будинку. Вихід в Інтернет забезпечує роутер, підключений з одного боку до локальної мережі а з іншого до мережі провайдера інтернету. Провайдер призначає роутера свій статичний або динамічний IP адреса і роутер здійснює трансляцію адрес локальної мережі в мережу провайдера. За замовчуванням правила цієї трансляції забезпечують вільну видимість інтернет-адрес з локальної мережі, але не дозволяють звернутися до локальних адресами з боку Інтернету. Є кілька способів обійти це обмеження [17].

Більшість сучасних роутерів дозволяють задати додаткові правила трансляції мережових адрес між локальної та глобальної мережами. Як правило для цього використовуються технології Virtual server або DMZ. Обидві технології дозволяють звернутися до сервера в локальній мережі з глобальної мережі, знаючи лише IP адреса, виданий роутера провайдером. У разі статичного IP адреси роутера це часто може бути задовільним рішенням для обмеженого кола користувачів системи. Однак такий підхід не завжди зручний: необхідно вручну конфігурувати роутер і з'ясовувати IP-адреса роутера, який може регулярно змінюватися. Щодо легко вирішити проблему невідомого IP адреси можна за допомогою механізму DDNS [19].

Щоб полегшити проблему доступності пристрою в Інтернеті і зробити інсталяцію пристрою легкою для користувача були розроблений ряд рішень. Механізм цих рішень базується на існуванні в Інтернеті спеціального

сервера, до якого може підключитися як IoT пристрій, так і планшет / комп'ютер користувача. При цьому пристрій працює в режимі клієнта, ніяких спеціальних налаштувань роутера або особливих навичок від інстальатора і користувача пристрою не потрібно. Обмін даними з пристроєм здійснюється за посередництва цього спеціального сервісу, параметри якого в пристрій повинен закласти розробник. Поширення використання таких сервісів стримується необхідністю тривалий час підтримувати свій сервіс в Інтернеті або користуватися чужими сервісами з незрозумілими перспективами тривалого існування безкоштовних можливостей або регулярною оплатою комерційних варіантів [18].

2.3 Сервер на основі ESP8266

Щоб підключити ESP8266 до программатору FTDI, потрібні такі контакти (рис.2.1) [19]:

- контакт RX на ESP8266 – до контакту TX на программаторі.
- контакт TX - до контакту RX.
- контакт CH_PD - до контакту 3.3V.
- контакт GPIO 0 - до контакту GND.
- контакт VCC - до контакту 3.3V.
- контакт GND - до контакту GND.
-

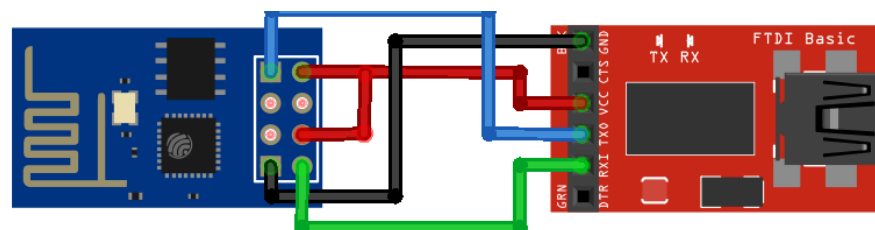


Рис.2.1. – Схема підключення ESP8266 до программатору FTDI

Підключивши ESP8266 до программатору FTDI, завантажте флешер

NodeMCU (це файл формату «* .exe»). Відкрийте флешер, який тільки що завантажили. Повинно відкритися приблизно ось таке вікно (рис.2.2):

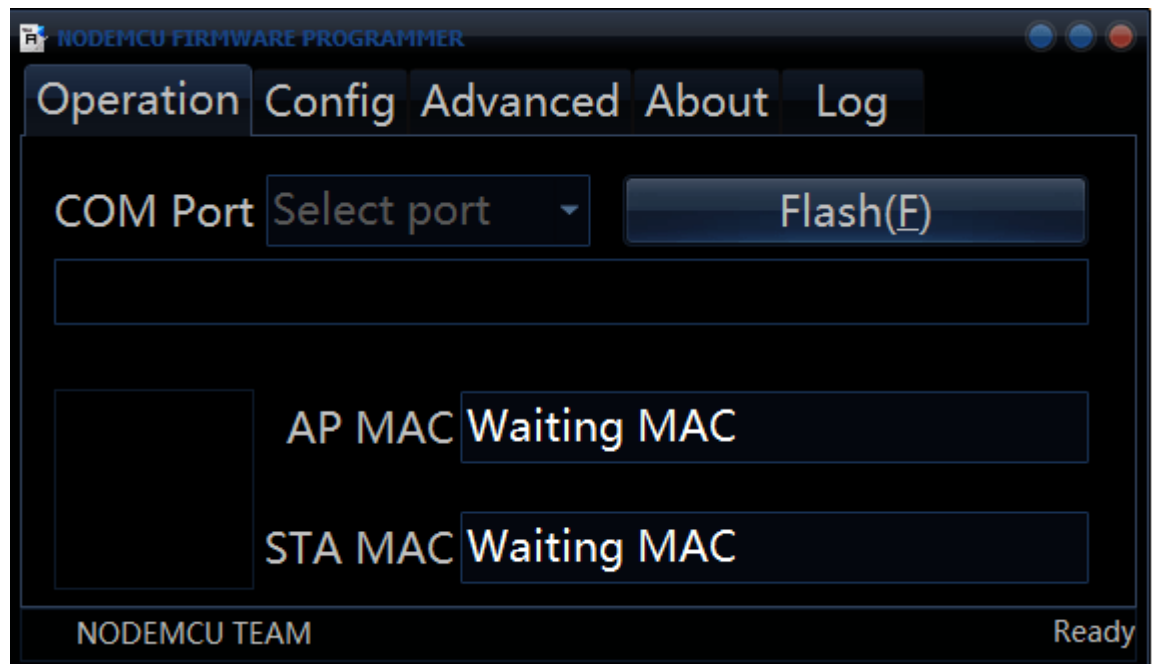


Рис.2.2. – Флешер для програмування контролера ESP8266

Зверніть увагу на вкладку «Advanced» - тут знаходяться додаткові настройки; якщо потрібно, поміняйте тут що-небудь. Потім поверніться на вкладку «Operation» і натисніть на кнопку «Flash». В ту ж секунду повинен початися процес завантаження прошивки. Якщо все завантажиться правильно, в лівому нижньому кутку флешерів з'явиться кругла зелена іконка з галочкою всередині [19].

Для цього проекту вам знадобляться, крім модуля ESP8266, джерело живлення (наприклад, батарея) і два світло діода (рис.2.3). Для завантаження LUA-скриптів на ESP8266 рекомендовано використовувати програму ESPlorer IDE (рис.2.4) [20].

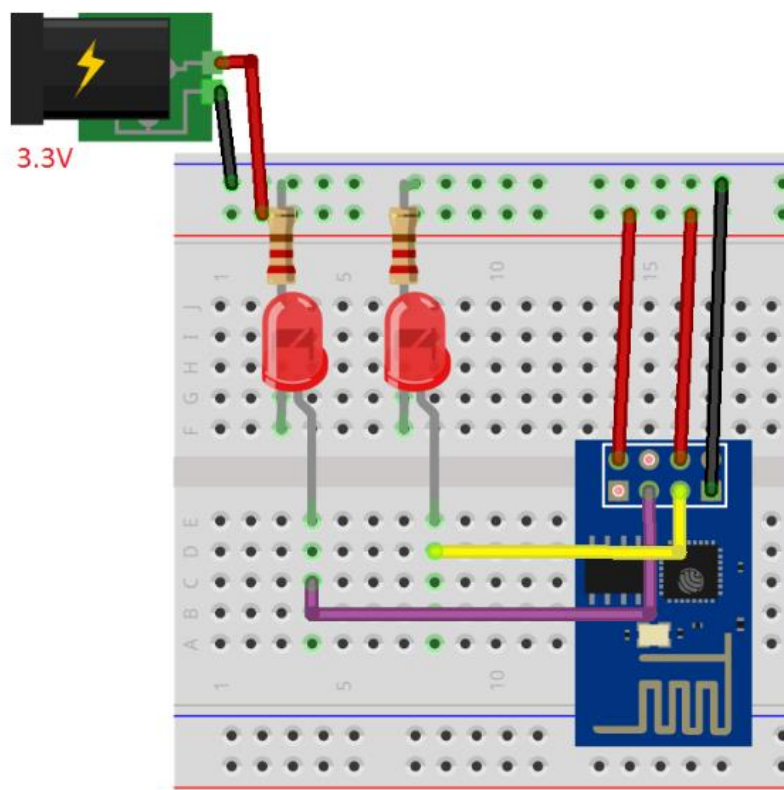


Рис.2.3. – Схема проекту із використанням контролера ESP-8266

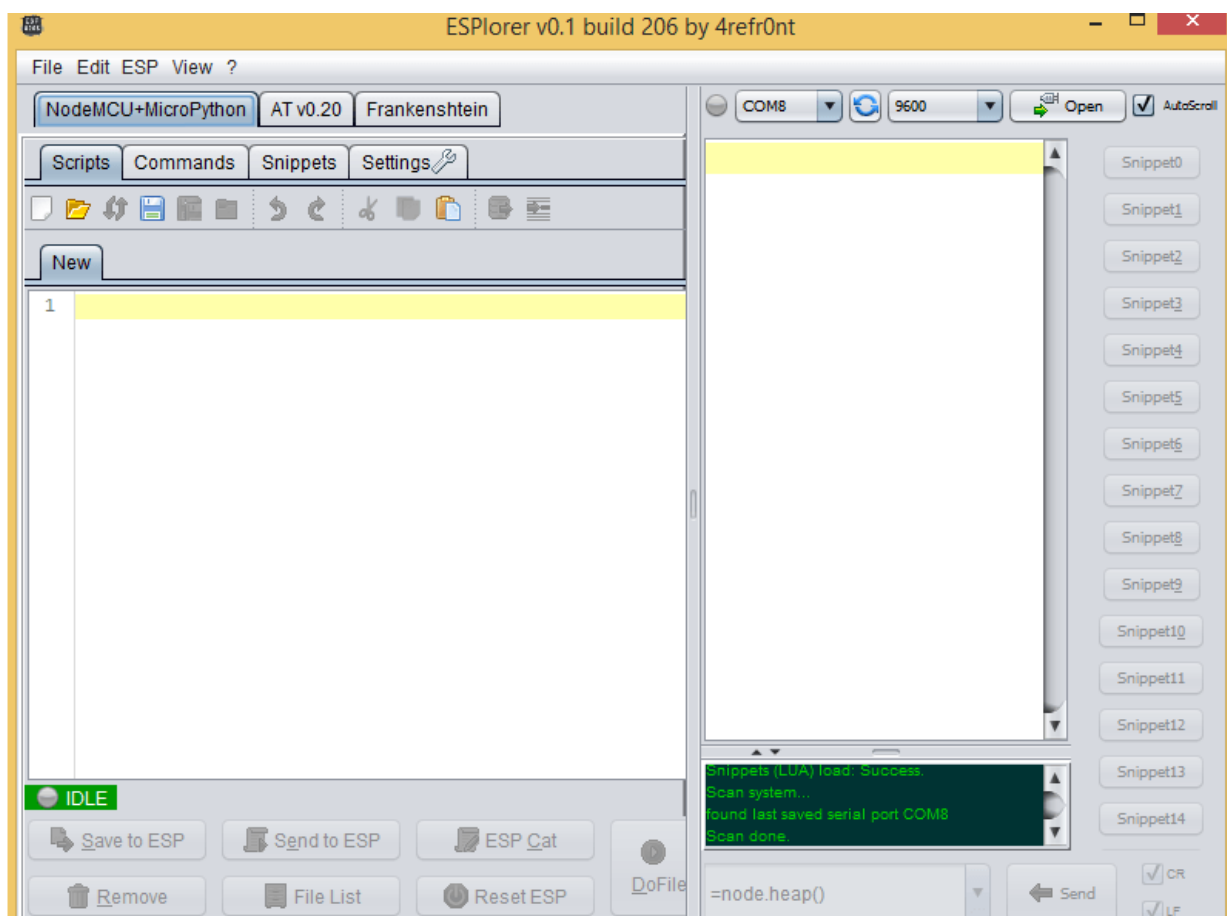


Рис.2.4. – Інтерфейс програми ESPlorer IDE

Тепер настала черга для завантаження LUA-скрипта. Для цього виконайте наступне:

- Підключіть програматор FTDI до комп'ютера.
- Виберіть порт, до якого підключений програматор.
- Натисніть кнопку «Open / Close».
- Виберіть вкладку «NodeMCU + MicroPython».
- Створіть новий файл під назвою «init.lua».
- Натисніть кнопку «Save to ESP».
- Всі елементи програми, необхідні для цих дій, виділені на зображенні нижче червоним кольором (рис.2.5).

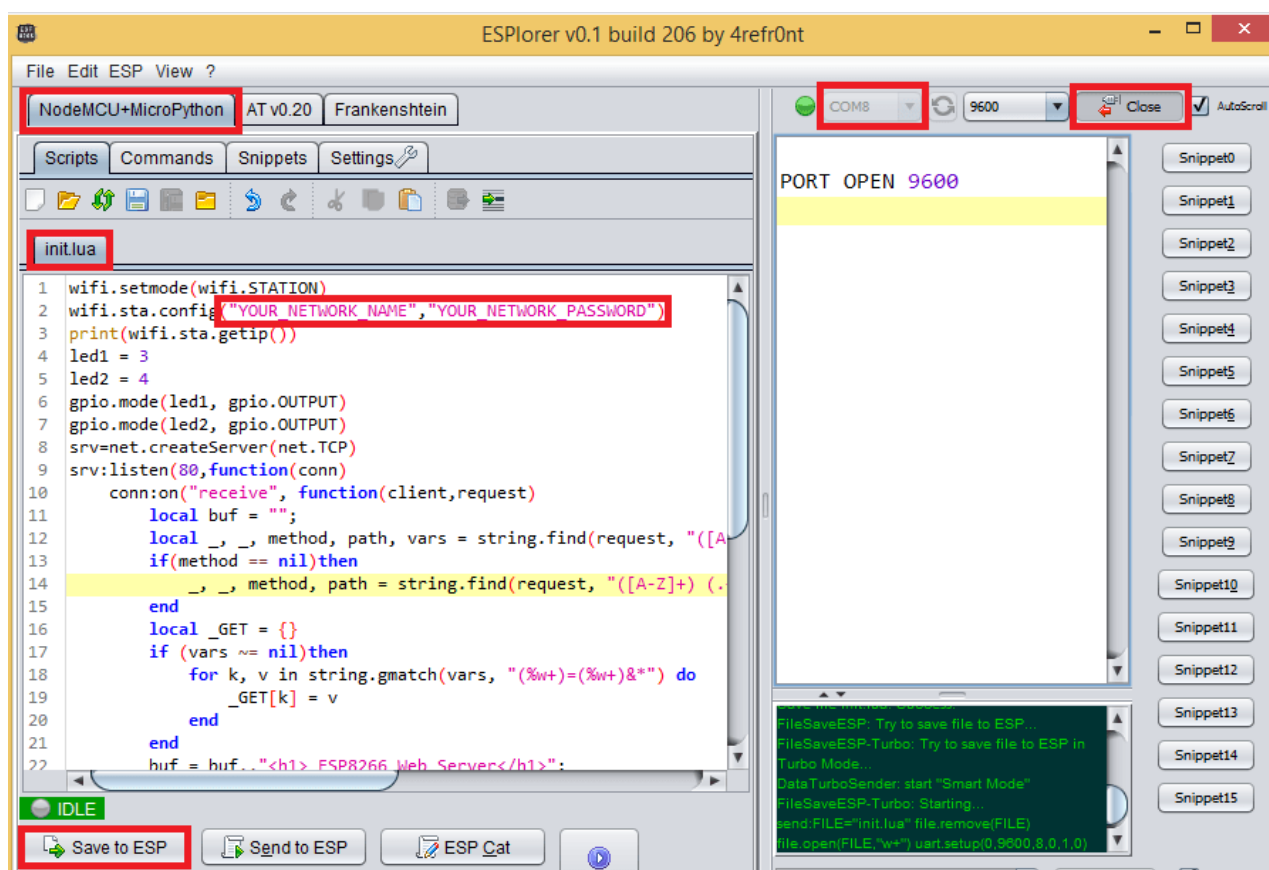


Рис.2.5. – Необхідні елементи програми

Тепер за допомогою ESPlorer IDE завантажити до ESP8266 скрипт, показаний нижче. Файл повинен бути названий «init.lua».


```

wifi.setmode(wifi.STATION)

wifi.sta.config("YOUR_NETWORK_NAME", "YOUR_NETWORK_PASSW
ORD")
print(wifi.sta.getip())
led1 =3
led2 =4
gpio.mode(led1, gpio.OUTPUT)
gpio.mode(led2, gpio.OUTPUT)
srv=net.createServer(net.TCP)
srv:listen(80, function(conn)
    conn:on("receive", function(client, request)
        local buf=""
        local _, _, method, path,
vars=string.find(request, "([A-Z]+) (.+)?(.+) HTTP");
        if(method ==nil) then
            _, _, method, path =string.find(request,
"([A-Z]+) (.+) HTTP");
        end
        local _GET = {}
        if (vars~=nil) then
            for k, v instring.gmatch(vars,
"(%w+)=(%w+)&*" ) do
                _GET[k] = v
            end
        end
        buf=buf.."<h1> ESP8266 Web Server</h1>";
        buf=buf.."<p>GPIO0 <a
href=\"?pin=ON1\"><button>ON</button></a>&nbsp;<a
href=\"?pin=OFF1\"><button>OFF</button></a></p>";
        buf=buf.."<p>GPIO2 <a
href=\"?pin=ON2\"><button>ON</button></a>&nbsp;<a
href=\"?pin=OFF2\"><button>OFF</button></a></p>";
        local _on, _off="", ""
        if(_GET.pin=="ON1") then
            gpio.write(led1, gpio.HIGH);
        elseif(_GET.pin=="OFF1") then
            gpio.write(led1, gpio.LOW);
        elseif(_GET.pin=="ON2") then
            gpio.write(led2, gpio.HIGH);
        elseif(_GET.pin=="OFF2") then
            gpio.write(led2, gpio.LOW);
        end
    end
end

```

```

        client:send(buf);
        client:close();
        collectgarbage();
    end)
end)

```

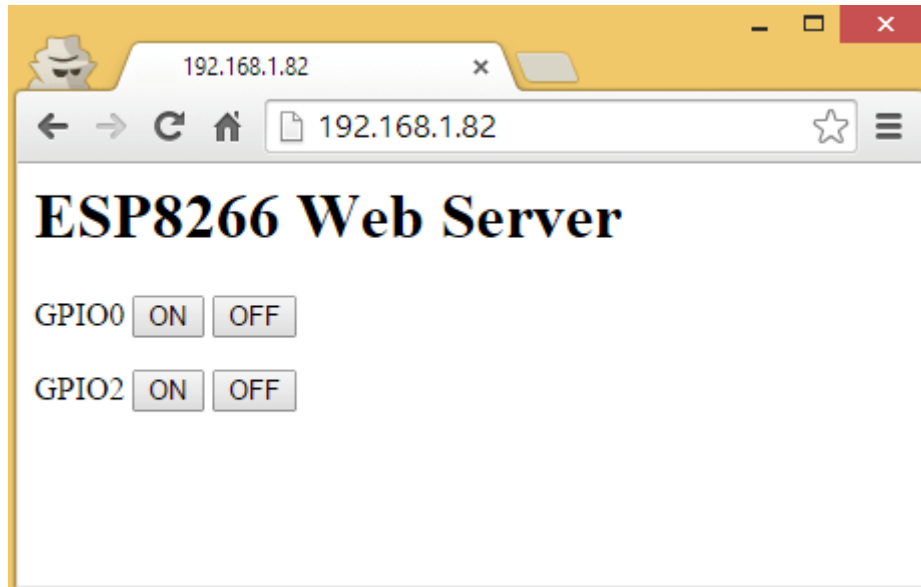


Рис.2.6. – Налаштований веб-сервер

Запустивши знову, ESP8266 надрукує в моніторі порту ESPlorer IDE власний IP-адреса. У моєму випадку це «192.168.1.82». Необхідно вписати цю IP-адресу в адресний рядок браузера і натиснути Enter. В результаті, якщо все налаштовано правильно, ми повинні отримати доступ до веб-сервера, що дозволяє управляти двома світло діодами (рис.2.6).

2.4 Реалізація сервера на основі ESP-8266 із використанням Arduino IDE

Програмування мовою C++ більш просте, у порівнянні із низькорівневими мовами, створення веб контенту такими мовами ставить питання ще гостріше. Програмування мікроконтролерних платформ на базі ESP-8266 все частіше здійснюється із використанням середовища розробки Arduino IDE, це зумовлено тим, що перехід на апаратну платформу ESP

здійснюється, все частіше, користувачами Arduino, які хочуть керувати апаратним забезпеченням через мережу інтернет. Також для Arduino IDE розроблено велику кількість бібліотек для роботи з різноманітними датчиками.

Розглянемо реалізацію Web-сервера програмними засобами Arduino IDE із інтегрованими бібліотеками для платформи ESP-8266. Нами було розроблено систему вимірювання аналогового сигналу, з подальшим відображенням результатів вимірювання на Web-сторінці, доступній в мережі інтернет.

Налаштування веб-сервера на ESP8266 в Arduino IDE забезпечується завдяки функціоналу універсальної бібліотеки `ESP8266WiFi.h`.

Першим етапом розробки є створення об'єкта сервера:

```
WiFiServer server(80);
```

Сервер відповідає клієнтам (в даному випадку - Web-браузерам) на порту 80, який є стандартним для спілкується з веб-серверами.

Далі опишемо функцію `prepareHtmlPage()`, яка поверне змінну типу `String`, що містить вміст веб-сторінки. Потім ми передамо цю змінну серверу, щоб передати її клієнту:

```
String prepareHtmlPage()
{
    String htmlPage =
        String("HTTP/1.1 200 OK\r\n") +
        "Content-Type: text/html\r\n" +
        "Connection: close\r\n" + // the
connection will be closed after completion of the
response
        "Refresh: 5\r\n" + // refresh the page
automatically every 5 sec
        "\r\n" +
        "<!DOCTYPE HTML>" +
        "<html>" +
        "Analog input:  " +
String(analogRead(A0)) +
        "</html>" +
        "\r\n";
```

```
    return htmlPage;
}
```

Дана функція збирає текстовий заголовок та вміст HTML сторінки.

Опишемо інформацію про те, який тип вмісту слід отримувати, та як він буде подаватися:

```
Content-Type: text/html
Connection: close
Refresh: 5
```

У цьому випадку тип вмісту є `text/html`, з'єднання буде закрито після подачі, і клієнт повинен запитувати контент знову кожні 5 секунд. Заголовок закінчується порожнім рядком `\r\n` (перехід на новий рядок та закінчення поточної). Це потрібно, щоб відрізнити заголовок від вмісту, який слід.

```
<!DOCTYPE HTML>
<html>
Analog input: [Value]
</html>
```

Вміст містить два основних теги HTML, один для позначення типу документа HTML, а інший для позначення початку та кінця документа. Всередині знаходиться неочищене значення, яке зчитується з аналогового входу ESP, перетвореного до типу

```
<!DOCTYPE HTML><html></html>analogRead(A0)String
String(analogRead(A0))
```

Обслуговування цієї веб-сторінки буде здійснюватися у циклі `loop()`, де сервер чекає, коли підключиться новий клієнт та надішле запит:

```
void loop()
{
    WiFiClient client = server.available();
    if (client)
    {
        // we have a new client sending some request
    }
}
```

Після підключення клієнта сервер прочитає його запит і транслює його до

послідовного порту:

```
while (client.connected())
{
    if (client.available())
    {
        String line = client.readStringUntil('\r');
        Serial.print(line);
    }
}
```

Запит від клієнта позначається порожнім рядком. Якщо сервер знаходить цю мітку, ми можемо відправити веб-сторінку та цикл `while()` із подальшим виходом за допомогою оператора `break`.

```
if (line.length() == 1 && line[0] == '\n')
{
    client.println(prepareHtmlPage());
    break;
}
```

Весь процес завершується припиненням зв'язку з клієнтом:

```
client.stop();
```

Взаємодія клієнта із сервером здійснюється через браузер із зазначенням IP адреси сервера, також, доступ може здійснюватись за адресою сайту, якщо серверу призначено DNS-адресу. При підключення виконується оновлення ssid та password, щоб отримати облікові дані точки доступу. Спочатку слід побачити підтвердження того, що модуль підключений до точки доступу та веб-сервера:

```
Connecting to sensor-net ..... connected
Web server started, open 192.168.1.104 in a web
browser
```

Для отримання даних із сервера, потрібно ввести надану ним IP-адресу у веб-браузері, після чого буде відображено сторінку, яку обслуговує ESP8266 (рис. 2.7).

Сторінка оновлюватиметься кожні 5 секунд, щоразу, коли це відбувається буде відображено запит від клієнта (веб-браузера) у послідовному порту:

```
[Client connected]
```

```

GET / HTTP/1.1
Accept: text/html, application/xhtml+xml, */*
Accept-Language: en-US
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64;
Trident/7.0; rv:11.0) like Gecko
Accept-Encoding: gzip, deflate
Host: 192.168.1.104
DNT: 1
Connection: Keep-Alive
[client disconnected]

```

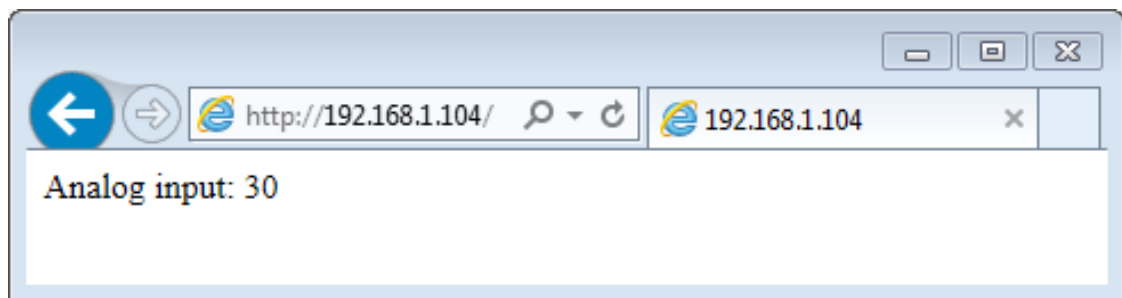


Рис.2.7. – Сторінка з даними, переданими Web-сервером

Протокол запиту та відповіді починається із заголовка, який містить інформацію про спілкування. Він містить тип вмісту, котрий повідомляється або приймається як `text/html`. У ньому зазначено, збережеться з'єднання чи закритим після надсилання заголовка. Також він містить ідентифікацію відправника: `User-Agent: Mozilla/5.0 (Windows NT 6.1)`

Результати кваліфікаційної роботи були представлені і обговорені на Міжнародній науково-методичній конференції «Фізика, електроніка, електротехніка ФЕЕ-2020» [21].

ВИСНОВКИ

1. Показано, що система віддаленого моніторингу фізичних величин може бути побудована із використанням мікроконтролерних платформ, а web-інтерфейс можна реалізувати із використанням мов гіпертекстової розмітки безпосередньо у пам'яті контролера. Розглянуто структуру, основні принципи функціонування веб-серверів та їх характеристики.

2. Вивчено характеристики, фізичні основи та принципи роботи датчиків фізичних величин, які можуть бути використані у системі моніторингу з підключенням до контролера ESP і передачі даних у мережу. Показано, що мікрокод контролера розміщується у пам'яті підключеного за протоколом SPI флеш-накопичувачі.

3. Установлено, що в пам'яті мікроконтролера ESP-8266 може бути розміщений код для реалізації базових функцій сервера і розмітка простої веб сторінки, функціонал якої може забезпечити функції моніторингу та керування віддаленим обладнанням.

4. Розроблено функціонал сервера для забезпечення керування живлення двох пристроїв, підключених до платформи Node-MCU на базі мікроконтролера ESP-8266. Реалізовано Web-сервер із використанням програмних засобів та бібліотек Arduino IDE для збору аналогових даних через мережу інтернет.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Schwartz M. Internet of Things with ESP8266. — Packt Publishing, 2016. — ISBN 9781786468024.
2. Скотт Хокинс. Администрирование веб-сервера Apache и руководство по электронной коммерции = Apache Web Server Administration and e-Commerce Handbook. – Москва: Вильямс, 2001. – 336 с. – ISBN 0-13-089873-2.
3. Станек Уильям Р. Internet Information Services (IIS) 7.0. Справочник администратора. – Санкт-Петербург: Русская редакция, 2009. – 528 с. – ISBN 978-5-7502-0383-3.
4. Адамс Крис. Администрирование сервера IIS 7. – Москва: Бином, 2010. – 362 с. – ISBN 978-5-9518-0367-2.
5. Датчики в системах збору даних і управління / Тесленко В.А // ПиКАД. – 2004. – № 2, 50-51
6. Датчики измерения влажности(гигрометры). Электронный ресурс. Режим доступа: <http://www.devicesearch.ru/article/datchiki-vlajnosti>. Дата доступа 04.04.2020
7. Arduino + датчик DHT11 (влажность и температура). Электронный ресурс. Режим доступа: <http://smart-home.te.ua/?s=DHT11&submit=Search&lang=uk>. Дата доступа 11.04.2020.
8. Интеллектуальный детектор газа/ Владимир Семёнов // Современная электроника. – 2007. – №9, 16-19
9. Датчик газа MQ-4 (углекислi газы, дим). Электронный ресурс. Режим доступа: http://ablogintheworld.com/cat_31/tov_65236.html/. Дата доступа 07.04.2020
10. Напівпровідникові датчики температури. Электронный ресурс. Режим доступа: <http://mcucpu.ru/index.php/pdevices/datchiki/108-poluprovodnikovye-datchiki-temperatury>. Дата доступа 04.04.2020

11. Петин В.А. Проекты с использованием контроллера Arduino. – БХВ-Петербург, 2014. – 400 с. – ISBN 9785977533379
12. Соммер У. Программирование микроконтроллерных плат Arduino/Freduino – Санкт-Петербург.: БХВ-Петербург, 2014, 398с.
13. обзор Arduino. Электронный ресурс. Режим доступа: <http://arduino.ua/ru/about/>. Дата доступа: 1.04.2020
14. STM32. Электронный ресурс. Режим доступа: <https://en.wikipedia.org/wiki/STM32>. Дата доступа: 04.04.2020
15. характеристики Arduino Nano. Электронный ресурс. Режим доступа: <https://www.arduino.cc/en/Main/ArduinoBoardNano> Дата доступа: 05.04.2020.
16. Датчики: Справочное пособие / В.М. Шарапов, Е.С. Полищук, Н.Д. Кошевой, Г.Г. Ишанин, И.Г. Минаев, А.С. Совлуков. – Москва: Техносфера, 2012. – 624 с.
17. Микроконтролер ESP8266. Электронный ресурс. Режим доступа: <https://esp8266.ru/>. Дата доступа: 04.04.2020
18. D. A. Aziz. Webserver Based Smart Monitoring System Using ESP8266 Node MCUModule // International Journal Of Scientific & Engineering Research. – 2018. – V. 10, № 6. – P. 801–807.
19. P. Srivastava, M. Bajaj, A. S. Rana. IOT based controlling of hybrid energy system using ESP8266 // IEEMA Engineer Infinite Conference. –2018. – DOI: 10.1109/ETECHNXT.2018.8385294
20. Q. Liu, X. Yang, L. Deng. An IBeacon-Based Location System for Smart Home Control // Sensors. – 2018. – V. 18. – P. 1897–1910
21. Ришкова К.О, Лобода В.С. Микроконтролерні модулі як складові Web систем / Матеріали Міжнародної науково-технічної конференції студентів та молодих вчених «Фізика, електроніка, електротехніка ФЕЕ-2020». – Суми: СумДУ, 2020. – С.85.